Library Book Management App

A library is implementing a mobile app for managing its book inventory. The app allows librarians to add new books, check book details, and generate reports on book statistics.
On the server side, at least the following details are maintained:

- Id: Internal identifier for the book. Integer value greater than zero.
- Title: The title of the book. A string of characters.
- Author: The author of the book. A string of characters.
- Genre: Genre of the book. A string of characters. Eg. "Fiction", "Non-Fiction", "Mystery", etc.
- Year: Year when the book was published. A four-digit integer.
- ISBN: International Standard Book Number. A string of characters.
- Availability: Current availability status of the book. A string of characters. Eg. "Available", "Checked Out", "Reserved", etc.

The application should provide the following features (available without restarting the app):

● Registration Section (Separate Activity/Screen):

A. **(1p)**(0.5p) Add a Book: Add a new book using a **POST /book** call by specifying all book details. Available online and offline.

B. **(2p)**(1p) View all Books: View all books in the library using a **GET /all** call. The list should include Id, Title, Author, and Availability. In offline mode, an offline message and retry option should be provided. The data should persist on the device after retrieval, regardless of online, offline, or restart conditions. Upon successful retrieval, since the data is available on the device, additional server calls should not be performed.

C. **(1p)**(1p) View Book Details: By selecting a book from the list, the user can view all the details. Using **GET /book** call with the book Id, the data should be retrieved from the server each time and made available on the device.

C. (0.5p) Delete a Book: Delete a book using a **DELETE /book** call by sending the book Id. The action should be available in the details screen.

● Report Section (Separate Activity/Screen) - Available Online Only:

A. **(1p)**(0.5p) View Genres: View all available book genres in a list using a **GET /genres** call.

B. **(1p)**(0.5p) View Books by Genre: View all available books from the selected genre in a list using the **GET /books** call, by specifying the selected genre.

C. **(1p)**(0.5p) Top 10 Books: View the top 10 books in a list with details such as Title, Author, and Availability. Use the **GET /all** call and present the result in descending order by publication year.

D. (0.5p) Top 5 Authors: View the top 5 authors in a list with the author's name and the number of books. Use the same **GET /all** call and present the result in descending order by the number of books.

● Owner Section (Separate Activity/Screen) - Available Online Only:

A. (1p) Record Author's Name: Record the author’s name in application settings. Persisted to survive app restarts.

B. (1p) View Author's Books: View all books by the persisted author in a list showing book title, author, and genre. Use **GET /author** call by specifying the author’s name.

● **(1p)**(1p) On the server side, once a new book is added to the system, the server will send, using a WebSocket channel, a message to all the connected clients/applications with the new object. Each application that is connected will display the received object fields, in a human form

(not JSON text or toString) using an in-app “notification” (e.g., using a snack bar, toast, or an on-screen dialog).
(0.5p)(0.5p) During all server or database operations, a progress indicator will be displayed.
(0.5p)(0.5p) On all server or DB interactions, if an error message is received, the app should display the error message using a toast or snackbar. A log message should be recorded for all interactions (server or DB calls).

NOTE: If your laboratory grade is at least 4.5, only the bold points will be used to compute the exam grade.